IN THE CLAIMS:

Claims 1 - 21, 27, and 34 have been cancelled. Claims 36 - 41 have been added. Claims 22, 23, 25, 28, 30, 31, 32, and 35 have been amended, as follows:

Claims 1 - 21 (cancelled).

22. (currently amended) A method of verifying authenticity of a hardware element, comprising:

creating a digital signature of a hardware address in a data processing device of the hardware element, the hardware element being located in the data processing device;

storing the digital signature of the hardware address of the hardware element in a memory element located in the data processing device;

comparing the digital signature of the hardware element to a known value; and

~~permitting access~~ loading a device driver onto the hardware element only if the digital signature of the hardware element is same as the known value.

23. (currently amended) The method according to claim 22, wherein the method further includes decrypting the digital signature of the hardware element that is stored in the memory element before the digital signature is compared to the known value.

24. (original) The method according to claim 22, wherein the method further includes storing the hardware address of the hardware element in the memory element

25. (currently amended) The method according to claim 22, wherein the method further includes manipulating the hardware address of the hardware element that is stored in memory with a hash algorithm to generate the known value which is compared to the digital signature of the hardware element.

26. (original) The method according to claim 22, wherein the hardware element is a network adapter.

Claim 27 (cancelled).

28. (currently amended) The method according to claim 22, wherein creating the digital signature includes manipulating the hardware address of the hardware element with a hash algorithm and encrypting the hashed hardware address of the hardware element with a private key before the encrypted hashed key hardware address is stored in the memory element.

29. (currently amended) A hardware authenticity verification system, comprising:

a machine-readable storage medium; and

machine-readable program code, stored on the machine-readable storage medium, the machine-readable program code having instructions, which when executed cause a data processing device to:

create a digital signature of a hardware address of [[the]] a hardware element installed in the data processing device;

store the digital signature of the hardware address of the hardware element in a memory element;

compare the digital signature of the hardware element to a known value; and

~~permit access~~ load device driver software onto to the hardware element only if the digital signature of the hardware element is the same as the known value.

30. (currently amended) The hardware authenticity verification system according to claim 29, wherein the machine-readable program code ~~further~~ includes instructions, which when executed cause the data processing device to decrypt the digital signature of the hardware element that is stored in the memory element before comparing the digital signal of the hardware element to a known value.

31. (currently amended) The hardware authenticity verification system according to claim 29, wherein the machine-readable program code ~~further~~ includes instructions, which when executed cause the data processing device to store the hardware address of the hardware element in the memory element.

32. (currently amended) The hardware authenticity verification system according to claim [[29]] 31, wherein the machine-readable program code ~~further~~ includes instructions, which when executed cause the data processing device to manipulate the hardware address of the hardware element that is stored in memory with a hash algorithm to generate the known value which is compared to the digital signature of the hardware element.

33. (original) The hardware authenticity verification system according to claim 29, wherein the hardware element is a network adapter.

Claim 34 (cancelled).

35. (currently amended) The hardware authenticity verification system according to claim 29, wherein the machine-readable program code ~~further~~ includes instructions, which when executed cause the data processing device to manipulate the

hardware address of the hardware element with a hash algorithm and encrypt the hashed hardware address of the hardware element with a private key before the encrypted hashed key hardware address is stored in the memory element.

36. (new) A verification system, comprising:

a machine-readable storage medium; and

machine-readable program code, stored on the machine-readable storage medium, the machine-readable program code having instructions, which when executed cause a data processing device to:

create a digital signature of a hardware address of a network adapter installed in the data processing device;

store the digital signature of the hardware address of the network adapter in a memory element;

compare the digital signature of the network adapter to a known value; and

load device driver software onto the network adapter only if the digital signature of the network adapter is the same as the known value.

37. (new) The verification system according to claim 36, wherein the machine-readable program code includes instructions, which when executed cause the data processing device to decrypt the digital signature of the network adapter that is stored in the memory element before comparing the digital signal of the network adapter to a known value.

38. (new) The verification system according to claim 36, wherein the machine-readable program code includes instructions which when executed cause the data processing device to manipulate the hardware address of the network adapter with a

hash algorithm and encrypt the hashed hardware address of the network adapter with a private key before the encrypted hashed key hardware address is stored in the memory element.

39. (new) A method of verifying authenticity, comprising:

creating a digital signature of a hardware address in a data processing device of a network adapter, the hardware element being located in the data processing device;

storing the digital signature of the hardware address of the network adapter in a memory element located in the data processing device;

comparing the digital signature of the network adapter to a known value; and

loading a device driver onto the network adapter only if the digital signature of the network adapter is same as the known value.

40. (new) The method according to claim 39, wherein the method further includes decrypting the digital signature of the network adapter that is stored in the memory element before the digital signature is compared to the known value.

41. (new) The method according to claim 39, wherein creating the digital signature includes manipulating the hardware address of the network adapter with a hash algorithm and encrypting the hashed hardware address of the network adaptor with a private key before the encrypted hashed key hardware address is stored in the memory element.